VehicleSDF: A 3D generative model for constrained engineering design via surrogate modeling

Hayata Morita **Kohei Shintani** **Chenyang Yuan** **Frank Permenter**

Toyota Research Institute
Cambridge, MA
firstname.lastname@tri.global

Abstract

A main challenge in mechanical design is to efficiently explore the design space while satisfying engineering constraints. This work explores the use of 3D generative models to explore the design space in the context of vehicle development, while estimating and enforcing engineering constraints. Specifically, we generate diverse 3D models of cars that meet a given set of geometric specifications, while also obtaining quick estimates of performance parameters such as aerodynamic drag. For this, we employ a data-driven approach (using the ShapeNet dataset) to train VehicleSDF, a DeepSDF based model that represents potential designs in a latent space witch can be decoded into a 3D model. We then train surrogate models to estimate engineering parameters from this latent space representation, enabling us to efficiently optimize latent vectors to match specifications. Our experiments show that we can generate diverse 3D models while matching the specified geometric parameters. Finally, we demonstrate that other performance parameters such as aerodynamic drag can be estimated in a differentiable pipeline.

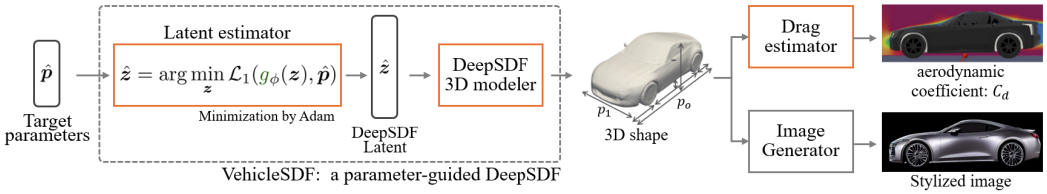
1 Introduction

Recent advances in generative AI have opened new possibilities for addressing mechanical design problems while considering both mechanical performance and aesthetics at the same time [1, 2]. Deep generative models have demonstrated remarkable capabilities in producing complex shapes and designs that satisfy multiple objectives simultaneously [3, 4]. Despite these advancements, the integration of engineering constraints into the generative design process remains a significant challenge. This research aims to bridge this gap by proposing a 3D generative model for vehicle design that simultaneously considers geometric constraints and aesthetic styling. Our proposed method is particularly relevant in the early stages of development, where rapid iterations and evaluations are crucial. By generating diverse 3D shapes that meet specific mechanical constraints while also producing aesthetically pleasing designs, the model has the potential to streamline the design process and reduce later revisions. The proposed pipeline (Figure 1a) consists of three key components:

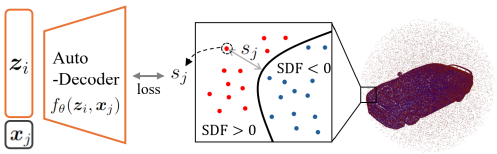
VehicleSDF: A 3D model generator To generate 3D vehicle shapes matching specified geometric parameters that come from engineering constraints (Figure 2), we train VehicleSDF, which is a differentiable parameter estimator combined with DeepSDF [5] trained on ShapeNet [6] dataset.

Drag coefficient predictor We want to efficiently estimate engineering parameters from generated models without running expensive simulations. To demonstrate this for vehicle drag coefficient (C_d), we trained a surrogate model [7, 1] achieving near-instant predictions from 3D models.

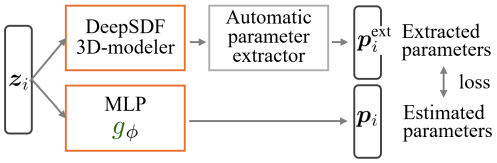
D3S3: Data-driven and Differentiable Simulations, Surrogates, and Solvers @ NeurIPS 2024.



(a) Proposed pipeline for vehicle design and performance estimation



(b) Training of DeepSDF using 3D shapes



(c) Training of surrogate parameter estimator

Figure 1: An illustration of our proposed pipeline and components. The latent vector z_i in (1b) is initialized randomly from $\mathcal{N}(0, \sigma^2)$ and optimized. The optimized and augmented latent vectors then are used in (1c). The orange-outlines mark the components we trained.

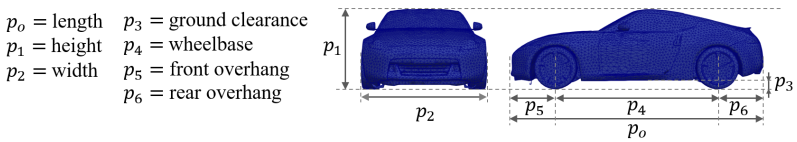


Figure 2: Vehicle geometric parameters to be specified during design

Image generator The final component demonstrates that we can generate photo-realistic vehicle renderings by using StableDiffusion with ControlNet [8] to stylize the 3D models.

By combining these components, VehicleSDF can generate diverse 3D models from vehicle geometric parameters. This enables designers to verify mechanical performance alongside aesthetic design during the early design phase. This integration can reduce the rework during later stage revisions.

2 Methodology

2.1 VehicleSDF: A Parameter-guided DeepSDF

We propose VehicleSDF, a parameter-guided DeepSDF for generating 3D shapes based on specified shape parameters. First, we train a DeepSDF model (Figure 1a) on 3D vehicle data (ShapeNet) to minimize a loss function that consists of the prediction error of the SDF values at each sample point, along with an L2 norm regularization term. For further details, refer to Appendix A. Second, to optimize the latent vectors to ensure conformity with the target parameters, we introduce a surrogate parameter estimator (Figure 1c). This estimator is a multi-layer perceptron $g_\phi(\boldsymbol{z}_i) = \boldsymbol{p}_i$, where $\boldsymbol{z}_i \in \mathbb{R}^m$ is the latent vector, $\boldsymbol{p}_i \in \mathbb{R}^n$ are the geometric parameters and ϕ are the model's weights, optimized at train-time:

$$\arg\min_\phi \sum_i \mathcal{L}_{\text{MSE}}(g_\phi(\boldsymbol{z}_i), \boldsymbol{p}_i^{\text{ext}}), \tag{1}$$

where $\boldsymbol{p}_i^{\text{ext}}$ are parameters extracted for each shape using extraction method of Appendix B and $\mathcal{L}_1$ is the mean-squared error. Fixing ϕ and minimizing $\mathcal{L}_{\text{MSE}}(g_\phi(\hat{\boldsymbol{z}}), \hat{\boldsymbol{p}})$, we can find latent vectors $\hat{\boldsymbol{z}}$ matching target parameters $\hat{\boldsymbol{p}}$.

2.2 A Surrogate Model for Drag Prediction

To predict C_d values, we trained a surrogate model using methods based on [7, 1] (Figure 3). Specifically, we first render the 3D object into an image that integrates normal maps from the top, bottom, left, right, front, and back views. Next, we numerically encode this image using a frozen feature extractor, and predict a C_d value from a trainable LightGBM [9] layer. Our model employs established techniques in transfer learning [10] to enhance robustness to out-of-distribution shifts.

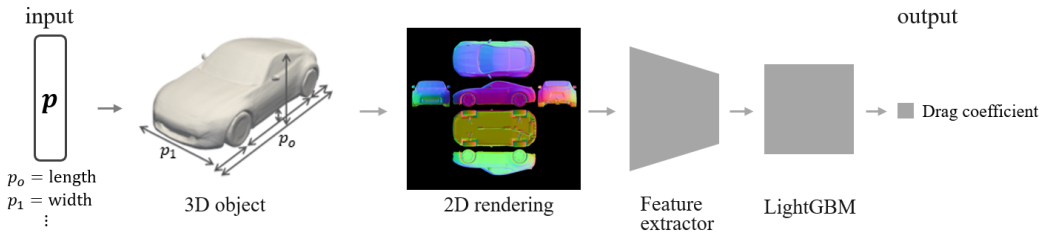


Figure 3: Drag estimation pipeline

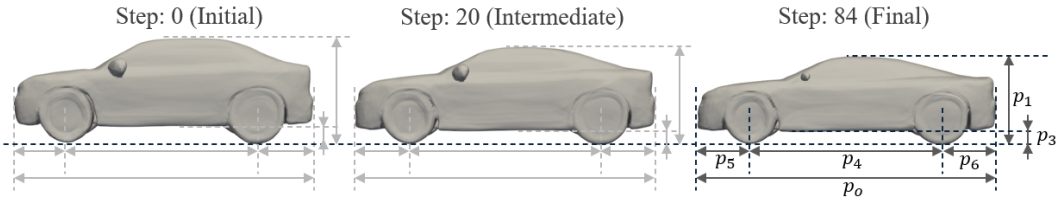


Figure 4: Optimization of 3D shape to match specified parameters

For training, we used the dataset [7] with the same train, validation, and test splits. The training results are discussed in Section 3.

2.3 Image Generation

To generate stylized vehicle renderings of the generated shapes, we adopted Stable Diffusion with ControlNet [8]. VehicleSDF generates 3D shapes, which are rendered into intermediate 2D images using either side-view normal map [11], depth map [12], or canny edges [13]. These rendered images are then used as inputs to ControlNet to generate stylized images.

3 Experiments

Data preparation and implementation To train VehicleSDF, we prepared the SDF sampling $\boldsymbol{X}_i := \{(\boldsymbol{x}_j, s_j) : s_j = \text{SDF}^i(\boldsymbol{x}_j)\}$ with 55,000 spatial points and 374 watertight shapes from ShapeNet, where $\text{SDF}^i(\boldsymbol{x}_j)$ represents the SDF value of spatial point $\boldsymbol{x}_j \in \mathbb{R}^3$ of shape i, calculated using [14]. Add reference to Appendix A. We also normalized each shape to fit within a unit sphere. To train the surrogate parameter estimator, we prepared a dataset including geometric parameters and the latent vector for each shape, further described in Appendix B. The parameters were normalized to unit length. To improve estimator accuracy, we also augmented the dataset using interpolated shapes generated by the trained VehicleSDF. An interpolated latent vector is defined as $\boldsymbol{r} = (1-\alpha)\boldsymbol{a} + \alpha\boldsymbol{b}$, where $\alpha \in (0, 1)$, $\boldsymbol{a}$ and $\boldsymbol{b}$ are optimized latent vector. We then augmented the training data from 374 to 10,000 shapes, after which the test MSE saturates. The final test MSE, train and test R^2 were 3.38×10^{-6}, 0.86 and 0.85, respectively.

Parametric 3D model generation with VehicleSDF We demonstrate that VehicleSDF generates 3D shapes that satisfy target geometric parameters. Figure 4 shows the initial, intermediate, and final shapes during optimization. Here, the initial shape refers to the shape reconstructed from a randomly

Table 1: Comparison of target geometric parameters and during optimization

Parameters	p_0	p_1	p_2	p_3	p_4	p_5	p_6	MSE
Initial	1.000	0.331	0.396	0.053	0.598	0.194	0.208	5.97×10^{-4}
Intermediate	1.000	0.306	0.425	0.039	0.599	0.203	0.199	1.03×10^{-4}
Final	1.000	0.280	0.431	0.037	0.600	0.200	0.200	2.86×10^{-8}
Target	1.000	0.280	0.430	0.037	0.600	0.200	0.200	-

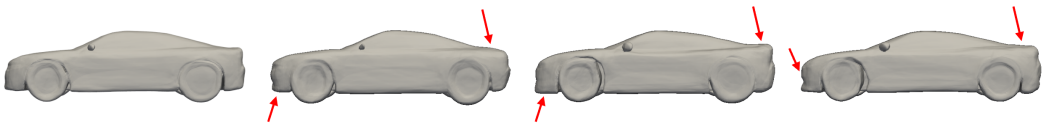


Figure 5: Results of optimization starting from four different initial shapes. The arrow indicate the points of change from the leftmost image. Figure 13 in Appendix E show another examples.

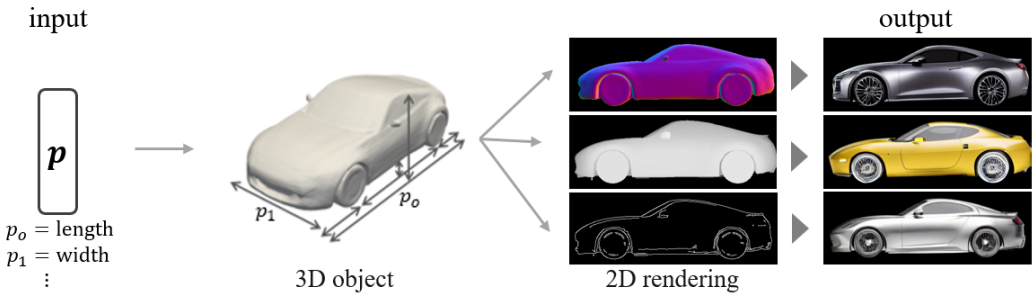


Figure 6: Generating styling images by ControlNet with the input of 3D shapes.

initialized latent vector from $\mathcal{N}(0, 0.01^2)$. The final shape closely matches the target parameters, as seen in Table 1.

Various 3D shape generation Geometric parameters and 3D shapes have a one-to-many relationship, where various shapes can satisfy a single set of target parameters. VehicleSDF can generate various final shapes from different initialized latent vectors. Figure 5 shows the results generated from four different initial latent vector by inputting the target parameters from Table 1. It can be observed that the rear and front silhouettes differ locally. This demonstrates that VehicleSDF can output 3D shapes that satisfy the geometric parameters with variations.

Drag coefficient prediction As the feature extractor for the drag coefficient prediction model shown in Figure 7, this research compares CLIP [15] and Vision Transformers (ViT) [16]. CLIP model embeds images into 512-dimensional vectors, while ViT model embeds them into 1024-dimensional vectors. We then employed a learnable LightGBM model to predict C_d values from these vectors. For CLIP features, the test R^2 and MSE were 0.55 and 0.0026, respectively, and for ViT features they were 0.58 and 0.0024. Since ViT features achieve a slightly better performance, we use them as part of our drag coefficient prediction model in VehicleSDF.

Image generation Figure 6 shows the results of generating detailed styling images using ControlNet [8] based on 2D renderings of the side views of 3D shapes generated by VehicleSDF, using normal map, depth map, and canny edge. These results demonstrate that the 3D shapes generated by VehicleSDF have sufficient quality for photo-realistic styling.

4 Conclusion, Limitations and Future Work

Generative AI tools that integrate engineering constraints, performance metrics, and aesthetics can accelerate the design process. This research demonstrates that we can modify current 3D shape generation techniques to incorporate engineering constraints. Additionally, we provided a proof of concept by generating C_d values and aesthetic vehicle renderings from the generated shapes. However, our approach does not currently allow direct optimization based on C_d values. Attempts to predict C_d from DeepSDF's latent space, similar to geometric parameters, were not successful as the latent space could not capture the shape information necessary for drag coefficient prediction. Future improvements could leverage larger datasets like DrivAerNet++ [17] and develop a fully differentiable pipeline from latent space to C_d prediction, which would be an exciting direction for future work.

References

[1] Nikos Arechiga, Frank Permenter, Binyang Song, and Chenyang Yuan. Drag-guided diffusion models for vehicle image generation. *arXiv preprint arXiv:2306.09935*, 2023.

[2] Kazunari Wada, Katsuyuki Suzuki, and Kazuo Yonekura. Physics-guided training of gan to improve accuracy in airfoil design synthesis. *Computer Methods in Applied Mechanics and Engineering*, 421:116746, 2024.

[3] Kazuo Yonekura and Katsuyuki Suzuki. Data-driven design exploration method using conditional variational autoencoder for airfoil design. *Structural and Multidisciplinary Optimization*, 64(2):613–624, 2021.

[4] WANG Jing, LI Runze, HE Cheng, CHEN Haixin, Ran Cheng, ZHAI Chen, and Miao Zhang. An inverse design method for supercritical airfoil based on conditional generative models. *Chinese Journal of Aeronautics*, 35(3):62–74, 2022.

[5] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.

[6] Angel X. Chang et al. Shapenet: An information-rich 3d model repository, 2015.

[7] Binyang Song, Chenyang Yuan, Frank Permenter, Nikos Arechiga, and Faez Ahmed. Surrogate modeling of car drag coefficient with depth and normal renderings. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 87301, page V03AT03A029. American Society of Mechanical Engineers, 2023.

[8] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

[9] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[10] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022.

[11] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z Dai, Andrea F Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R Walter, et al. Diode: A dense indoor and outdoor depth dataset. *arXiv preprint arXiv:1908.00463*, 2019.

[12] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020.

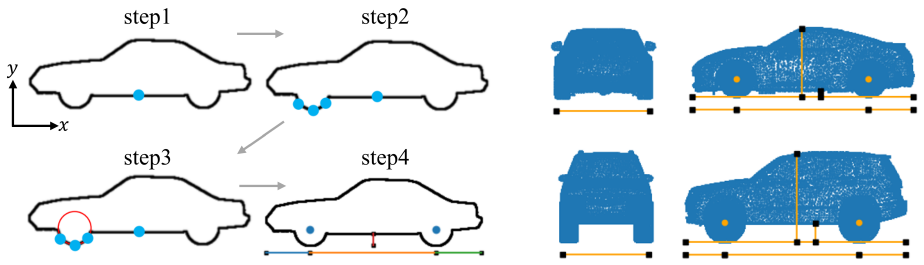
[13] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[14] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011. IEEE.

[15] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[17] Mohamed Elrefaie, Florin Morar, Angela Dai, and Faez Ahmed. Drivaernet++: A large-scale multimodal car dataset with computational fluid dynamics simulations and deep learning benchmarks, 2024.



(a) Automatic parameter extractor (b) Examples of extracted parameters

Figure 7: An example of extracting geometric parameters such as ground clearance, front and rear overhang, and wheelbase from ShapeNet shapes using method (a) and extraction method (b) is shown.

[18] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998.

A DeepSDF Training Details

DeepSDF considers a set of N shapes $\{\boldsymbol{X}_i\}$, where each shape $\boldsymbol{X}_i$ is sampled at K points $\boldsymbol{x}_j \in \mathbb{R}^3$, and each point is assigned an SDF value $s_j \in \mathbb{R}$. The SDF value s_j represents the signed distance from the surface, where points inside the surface have a negative sign, and points outside have a positive sign. The relationship between these K sampled points and the SDF values is expressed as: $\boldsymbol{X}_i := \{(\boldsymbol{x}_j, s_j) : s_j = \text{SDF}^i(\boldsymbol{x}_j)\}$. DeepSDF is trained by optimizing θ and $\boldsymbol{z}$ using the loss function:

$$\arg\min_{\theta, \boldsymbol{z}_i} \sum_{i=1}^{N} \left(\sum_{j=1}^{K} \mathcal{L}_{\text{MSE}} \left(f_\theta(\boldsymbol{z}_i, \boldsymbol{x}_j), s_j \right) + \frac{1}{\sigma_L^2} \|z_i\|_2^2 \right). \tag{2}$$

The first term in the equation is the loss function $\mathcal{L}_{\text{MSE}}$ applied to the output of the model $f_\theta : \mathbb{R}^m \times \mathbb{R}^3 \to \mathbb{R}$ and the truth s_j, and the second term is given by the L_2-norm of the latent $\boldsymbol{z}_i$ and $\sigma_L \in \mathbb{R}$ as regularization. Given a set of spatial points $\boldsymbol{x}_j$, a 3D shape is generated by evaluating $f_{\hat{\theta}}(\hat{\boldsymbol{z}}, \boldsymbol{x}_j)$ and extracting the isosurface where SDF = 0 using techniques such as ray casting or Marching Cubes [18]. Here, $\hat{\theta}$ and $\hat{\boldsymbol{z}}$ represent the optimized values of θ and $\boldsymbol{z}$, respectively.

B Automatic Parameter Extractor Details

We used four steps to extract geometric parameters from the 3D geometry, as shown in Figure 7a.

1. Extract the point in the vehicle with the central x-coordinate and the smallest y-coordinate.
2. Extract 3 points whose y-coordinate is smaller than the point obtained in Step 1 and whose x-coordinate is smaller than the center of the vehicle. These are the three points on the front tire.
3. Define a circle passing through all three points on the tire, and the center of the circle is the center of the front tire.
4. Likewise, the center of the rear tire was determined, and the ground clearance, wheelbase, front overhang, and rear overhang were extracted.

On the other hand, length and height were extracted from the difference between the largest and smallest points on the x-axis and y-axis, respectively. Width was defined from the difference of the maximum and minimum points in the z-direction among the points with the same y-coordinate as the tire center to avoid mirrors. An example of automatic measurement using this algorithm is shown in Figure 7b.

C Reconstruction and Interpolation Details

The results of the 3d models generated from VehicleSDF are shown in Figure 8, showing that it is possible to generate diverse vehicle shapes. Figure 9 shows the 3D models generated by interpolating between two designs. For interpolation, we used the interpolated latent $\boldsymbol{r}$ defined as $\boldsymbol{r} = \boldsymbol{a} + \alpha(\boldsymbol{b} - \boldsymbol{a})$, where $\alpha \in (0, 1)$, $\boldsymbol{a}$ and $\boldsymbol{b}$ are optimized latent, respectively. We can see from this that the vehicle shape can be continuously changed in between two arbitrary shapes. We also augment the training dataset for training the parameter estimator model with interpolated shapes.

D Surrogate Parameter Estimator Training

Figure 10 shows the relationship between the MSE of the held-out test set, defined at 20% of each data volume, with the amount of training data. The MSE saturated when the training data exceeded 10,000 examples. Figure 11 show the distribution of each parameter in the augmented data, as well as the shapes with the maximum and minimum values for each parameter (except *length* which is normalized to 1). The mean and variance of the augmented dataset {length, height, width, ground clearance, wheelbase, front overhang, rear overhang} were {1.00, 0.31, 0.40, 0.05, 0.60, 0.19, 0.21} and {0.0, 5.4×10^{-4}, 7.1×10^{-4}, 9.0×10^{-5}, 5.3×10^{-4}, 2.7×10^{-4}, 4.2×10^{-4}}, respectively.

E Validation of VehicleSDF

Figure 12 shows the result of generating 3D shapes with target parameters that are within the distribution of the training data used for VehicleSDF as well as parameters that are outside of that distribution. The results show that it is possible to generate shapes for target parameters outside the distribution (other than target (e)). However, we observe that the shape of target (i) was unnatural due to insufficient training data. For targets within the distribution, a shape closer to the target can be found by directly searching for the nearest neighbor shape using Euclidean distance in parameter space, while for targets outside the distribution other than target f, the results generated by the VehicleSDF were closer to the target than the nearest neighbors.

F Drag Coefficient Predictor Training Details

To train C_d value prediction model, we used a dataset of 9,070 shapes labeled with C_d values prepared in [7]. This dataset was augmented by a factor of 4 using width increments (2x) and flips (2x), so that only a quarter of the shapes are completely unique. The distribution of C_d values for this dataset is shown in Figure 14a. Also, we split the entire dataset into training, validation, and test sets following a ratio of 0.7:0.15:0.15 as in [7] and trained the prediction model. A comparison of the prediction with the ground truth is shown in Figure 14b.

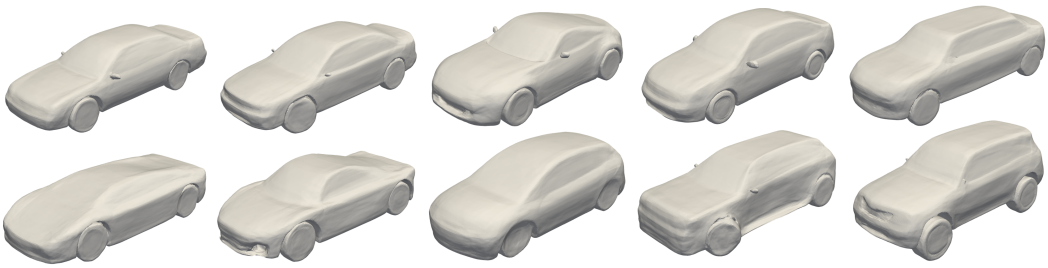


Figure 8: A illustration of diversity 3D shape generation by VehicleSDF.

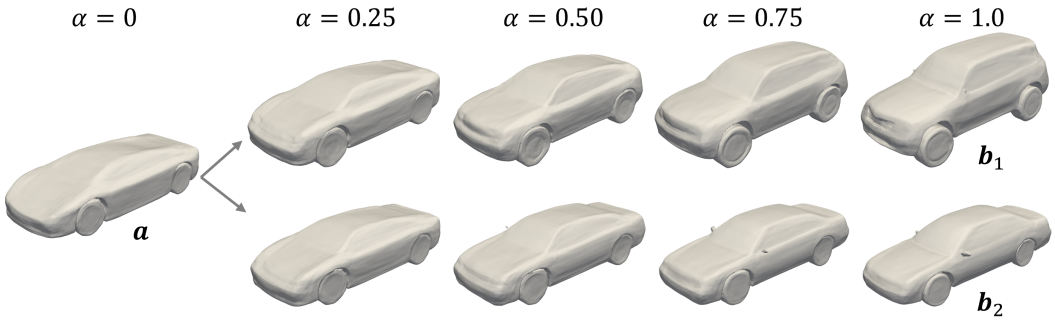


Figure 9: A illustration of interpolated generation from same initial shape to different target shapes

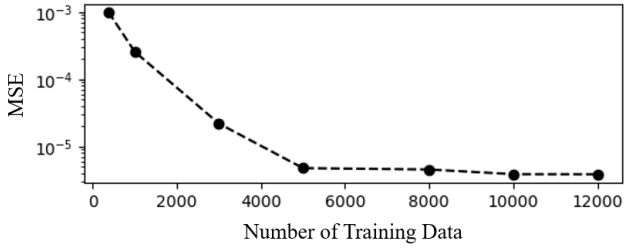


Figure 10: Shift in the parameter estimator's test MSE when augmenting dataset size from 374 to 12,000

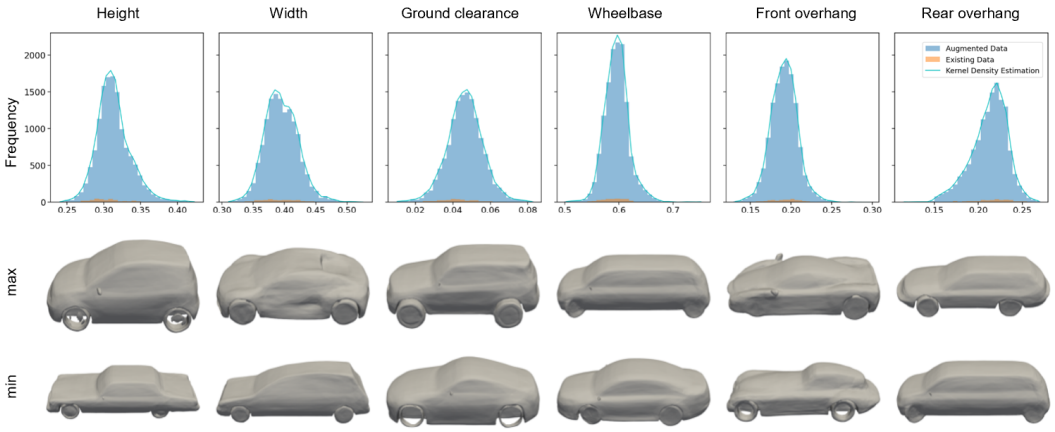
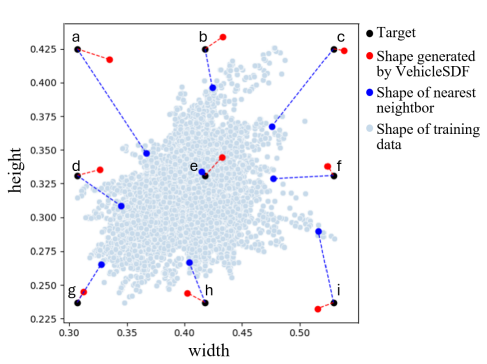
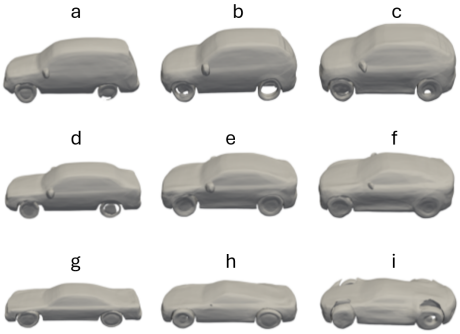


Figure 11: Distribution of the data used for the parameter estimation model. The top row shows the histograms of each parameter. The color differences indicate the existing data and the augmented data generated by the trained DeepSDF model. The middle row shows the 3D shape corresponding to the maximum value of each parameter, and the bottom row shows the 3D shape corresponding to the minimum value of each parameter.



(a) Target points and each shape parameters



(b) Shapes generated by VehicleSDF for each target

Figure 12: 3D shapes generated using VehicleSDF with nine target parameters are compared to nearest neighbor shapes in the training data to each target parameters (a). The nine target parameters consists of nine combinations of the maximum, minimum, and median values of height and width, plus the median values of the other parameters. The 3D shapes generated by VehicleSDF for each target is shown in (b).

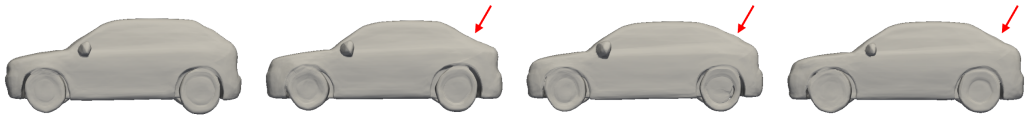
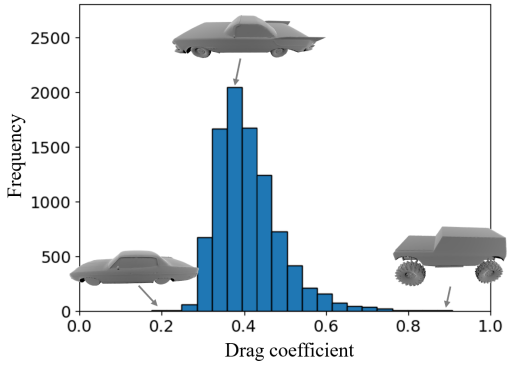
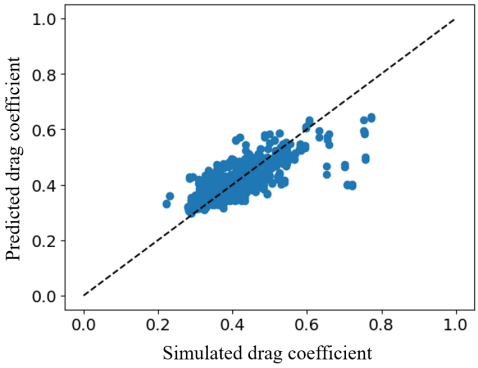


Figure 13: Another result for optimization starting from four different initial shapes. The arrow indicate the points of change from the leftmost image.



(a)



(b)

Figure 14: The drag coefficient (C_d) distribution for training drag predictor is shown in (a), and a comparison between predicted and ground-truth values of C_d is shown in (b).